

Exhibit 3

U.S. Patent No. 7,159,766 (“766 Patent”)**Accused Products**

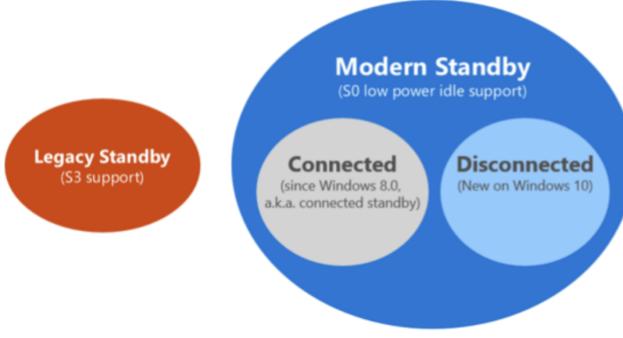
Samsung Wi-Fi enabled products including the Windows 10 operating system, including without limitation the Samsung Galaxy Book Flex 13.3" NP930QCG-K01US ("Accused Products") infringe at least Claims 1, 14, 45, and 57 of the '766 Patent.

Claim 1

Claim 1	Accused Products
[1pre] A system, comprising:	To the extent the preamble is limiting, each Accused Product comprises the claimed system. For example, the Galaxy Book Flex includes the Windows 10 operating system and practices the Windows 10 "Modern Standby" feature, as described below. <i>See, e.g.:</i>

Claim 1	Accused Products
	<div style="display: flex; align-items: center;"> <div style="flex: 1;">  <p>Windows 10 Home</p> </div> <div style="flex: 2;"> <p>Galaxy Book Flex, 13.3", 512GB, S Pen Included Wishlist</p> <p>NP930QCG-K01US ★★★★★ 3.5 (71) Share your product experience</p> <p>DISPLAY</p> <div style="display: flex; justify-content: space-around; font-size: small;"> 13.3" 15.6" +\$50 </div> <p>Free shipping and easy returns: Get it by Jul. 26 Ship to: 10001</p> <p>From \$19.79/mo for 48 mos at 0% APR® excludes tax and shipping</p> <p>WITH ELIGIBLE TRADE-IN® AND SAMSUNG FINANCING</p> <p>OR</p> <p>From \$949.99 \$1,549.99 Save \$400.00</p> <p>Earn 1% back in rewards</p> <p>BUY NOW</p> <p>Where to buy Free shipping on all orders</p> <p>Buying for your business? Go to Galaxy Book Flex, 13.3", 512GB, S Pen Included for Business ></p> <div style="display: flex; justify-content: space-around; margin-top: 10px;"> </div> </div> </div> <p data-bbox="766 1098 1886 1171">Screenshot from https://www.samsung.com/us/computing/galaxy-books/galaxy-book-flex/galaxy-book-flex-13-3-qled-512gb-storage-s-pen-included-np930qcg-k01us/</p>

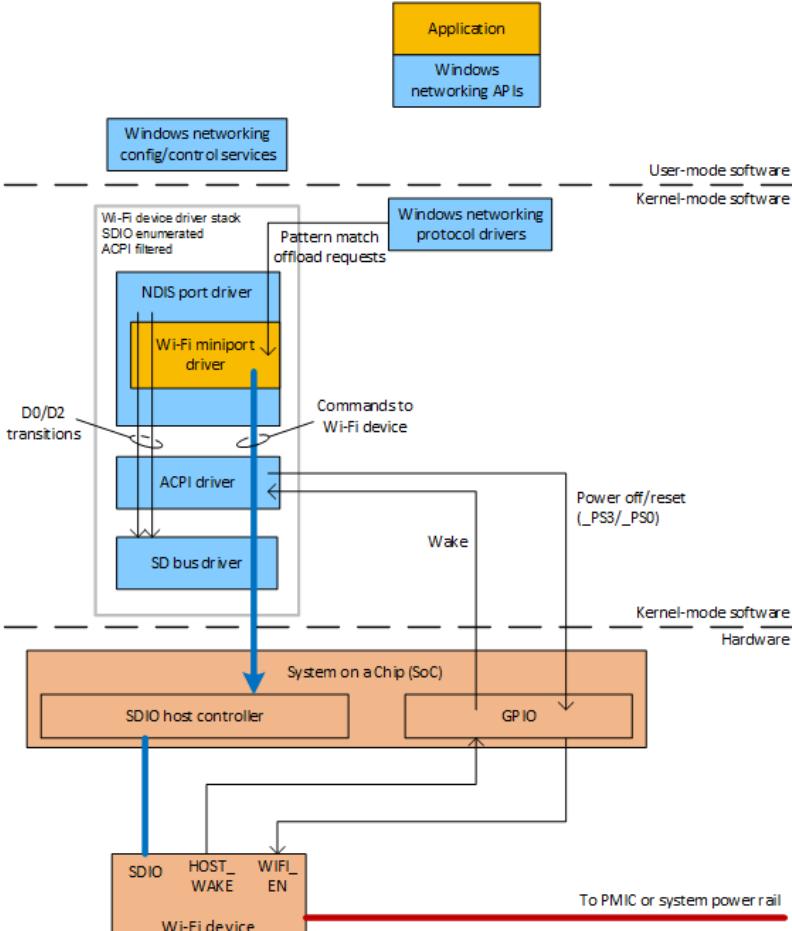
Claim 1	Accused Products
	<p>What is Modern Standby?</p> <p>05/02/2017 • 4 minutes to read •  +1</p> <p>Windows 10 Modern Standby (Modern Standby) expands the Windows 8.1 Connected Standby power model. Connected Standby, and consequently Modern Standby, enable an instant on / instant off user experience, similar to smartphone power models. Just like the phone, the S0 low power idle model enables the system to stay up-to-date whenever a suitable network is available.</p> <p>Although Modern Standby enables an instant on/off user experience like Connected Standby, Modern Standby is more inclusive than the Windows 8.1 Connected Standby power model. Modern Standby allows for market segments previously limited to the S3 power model to take advantage of the low power idle model. Example systems include systems based on rotational media and hybrid media (for example, SSD + HDD or SSHD) and/or a NIC that doesn't support all of the prior requirements for Connected Standby.</p> <p>The number of systems capable of S0 low power idle model is increasing and we expect more systems to use the always on, instantly available power model instead of the traditional S3/S4 power model. In the Modern Standby section, we have outlined important changes, partner requirements, and best practices for enabling Modern Standby.</p> <p> Note</p> <p>Modern Standby is available for both Windows 10 desktop and Windows 10 Mobile.</p> <p>https://docs.microsoft.com/en-us/windows-hardware/design/device-experiences/modern-standby?redirectedfrom=MSDN</p>

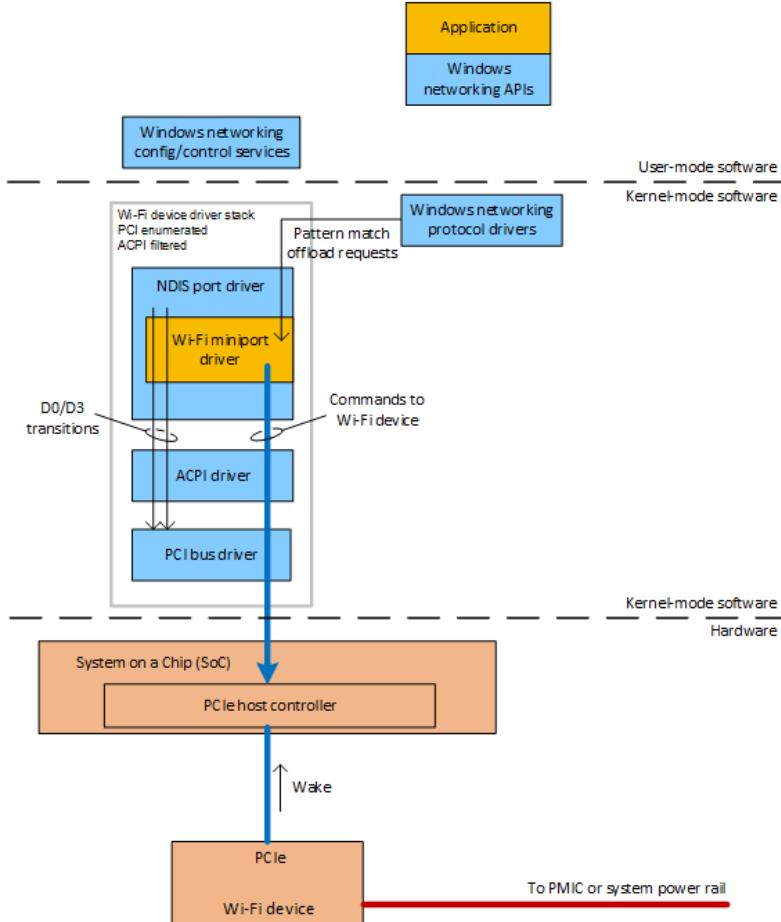
Claim 1	Accused Products
	 <p>The diagram illustrates the relationships between different power management models. On the left, a single orange circle represents Legacy Standby (\$3 support). On the right, a large blue circle represents Modern Standby (\$0 low power idle support), which is divided into two smaller circles: Connected (since Windows 8.0, a.k.a. connected standby) and Disconnected (New on Windows 10).</p> <p>The above figure illustrates the relationships between the models and how Modern Standby systems can either stay connected to or disconnect from the network while in standby.</p> <p>On any Modern Standby system, the system remains in S0 while in standby, allowing the following scenarios to work:</p> <ul style="list-style-type: none"> • Background activity • Faster resume from a low power state <p>On systems that can stay connected while in standby, wakes based on specific network patterns may also be set by the operating system to enable apps to receive the latest content such as incoming email, VoIP calls, or news articles.</p> <p>Resume from Modern Standby</p> <p>When the user causes the system to resume from standby, e.g. presses the power button, the display is immediately turned on and networking devices are restored to their normal, active operating modes. The time from the power button press to the turning on of the display is typically less than 500 milliseconds. After the display is turned on and the networking device returns to normal operating mode, desktop applications resume and the system returns to its normal, screen-on active behavior.</p>

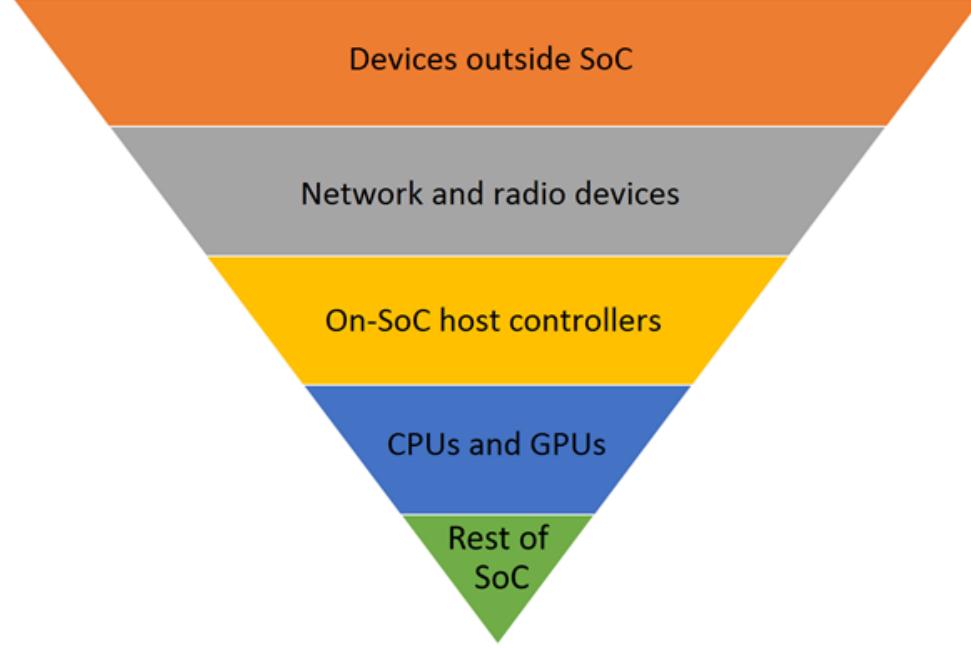
Claim 1	Accused Products
	<p>Activity during Modern Standby</p> <p>On-demand transitions to active mode can occur in response to user inputs, interrupts from networking devices and other hardware events. Windows transitions the SoC platform from active mode to idle mode after all software activity is stopped and the devices on and off the SoC chip have entered low-power states. (See Transitioning between active and idle states.)</p> <p>The networking and communications devices automatically transition between active and low-power modes, based on the software activity of the system during modern standby. When there are no system services or Microsoft Store app background tasks that require the network, the networking device is in the low-power, protocol offload, and WoL patterns mode. When a system service or background task requires network access, Windows automatically transitions the networking device to an active mode.</p> <p>Technical differences</p> <p>When in the lowest power state, systems may look very similar to systems in the S3 state—processors are powered off, memory is in self-refresh. The difference is in the path of how it enters and exits low power state. For S3 systems, the system is either active or in S3. For Modern Standby, the transition from the active to the low power state is a series of steps to lower power consumption. Components are powered down when they are not in use. So, the transition into and out of a lower power state is much quicker on a Modern Standby system than on an S3 system. This design also helps with the speed of entry and exit from Standby as it doesn't require firmware interactions.</p> <p>https://docs.microsoft.com/en-us/windows-hardware/design/device-experiences/modern-standby-vs-s3</p>

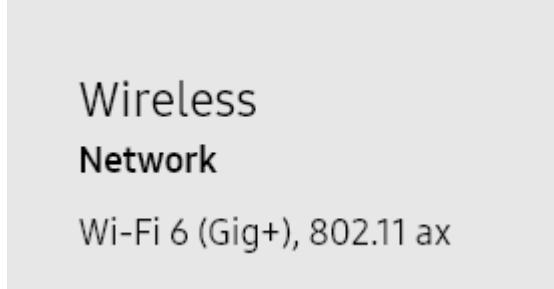
Claim 1	Accused Products
	<p>Transitioning from active to idle</p> <p>Windows transitions the SoC platform from active mode to idle mode after all software activity is stopped and the devices on and off the SoC chip have entered low-power states.</p> <p>The SoC transitions between idle and active modes during modern standby based on the triggers explained in the previous section. Whenever the SoC is active, Windows will aggressively attempt to return it to the idle (DRIPS) state to reduce power consumption.</p> <p>Note During modern standby, the SoC can be idle or active.</p> <p>The SoC will also be active when the system exits modern standby in response to user input, such as a power button press. The SoC will be put back in the idle (DRIPS) state automatically when all of the following conditions are true:</p> <ul style="list-style-type: none"> • All devices outside of the SoC have been powered down. • All network and radio devices have entered their low-power state to wait for interesting network packets or wake interrupts. • All host controllers on the SoC have been powered down. • All app background tasks have completed. • All CPU and GPU activity has stopped and all CPUs are idle. <p>Almost all modern standby power problems are related to making sure all of these five conditions have been met. Windows includes a built-in diagnostic tool called SleepStudy to help discover which of these conditions is not being met when the SoC is prevented from entering the idle (DRIPS) state.</p> <p>https://docs.microsoft.com/en-us/windows-hardware/design/device-experiences/transitioning-between-idle-and-active-states</p>
[1a] a processor;	<p>Each Accused Product includes a processor.</p> <p>For example, the Galaxy Book Flex includes an Intel Core i7-1065G7 processor.</p> <p><i>See, e.g.:</i></p>

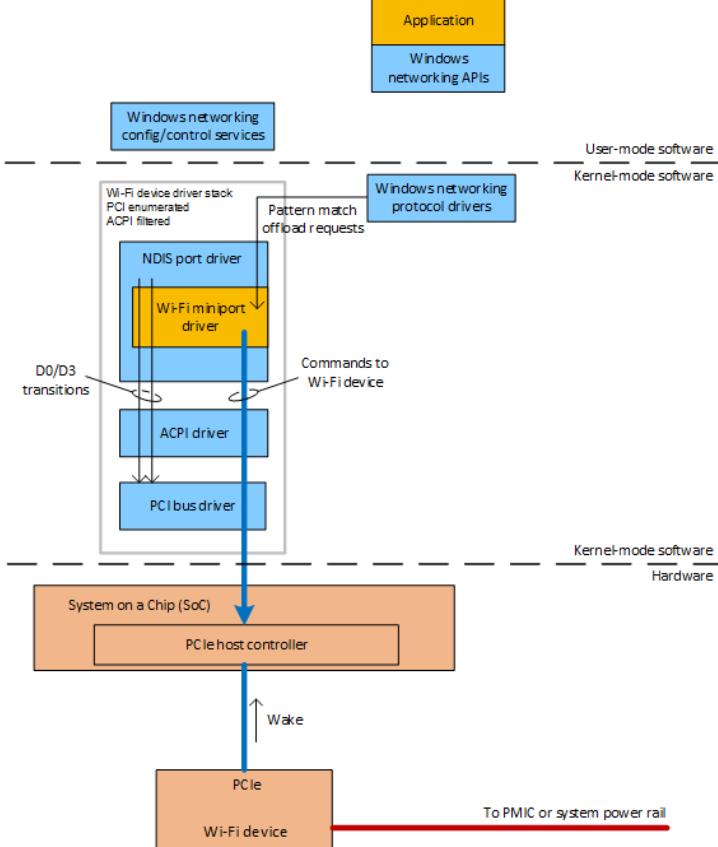
Claim 1	Accused Products
	<p>Processor</p> <p>Processor / Chipset</p> <p>Intel® Core™ i7-1065G7 Processor</p> <p>Screenshot identifying Intel Core i7-1065G7 processor from https://www.samsung.com/us/computing/galaxy-books/galaxy-book-flex/galaxy-book-flex-13-3-qled-512gb-storage-s-pen-included-np930qcg-k01us/</p>
<p>[1b] a host controller coupled to the processor; and</p>	<p>Each Accused Product comprises a host controller coupled to the processor.</p> <p>For example, the gram 15 comprises a host controller within the system-on-a-chip (SoC) for communicating with other devices. For one example, any device supporting Windows 10 Modern Standby must include either a PCIe host controller, an SDIO host controller, or a proprietary SoC bus controller.</p> <p><i>See, e.g.:</i></p>

Claim 1	Accused Products
	 <p>The diagram illustrates the Windows networking architecture, divided into User-mode software, Kernel-mode software, and Hardware layers.</p> <p>User-mode software: Application (Windows networking APIs) interacts with Windows networking config/control services.</p> <p>Kernel-mode software: Windows networking protocol drivers handle pattern match offload requests. The Wi-Fi device driver stack (SDIO enumerated, ACPI filtered) contains the NDIS port driver and Wi-Fi miniport driver. The ACPI driver manages D0/D2 transitions and sends commands to the Wi-Fi device. The SD bus driver is also shown.</p> <p>Hardware: The System on a Chip (SoC) contains the SDIO host controller and GPIO. The Wi-Fi device (SDIO, HOST_WAKE, WIFI_EN) is connected to the SoC and a power source (To PMIC or system power rail).</p> <p>Legend:</p> <ul style="list-style-type: none"> Windows components (blue) Partner components (yellow) Hardware components (orange) <p>Data communication: Indicated by blue arrows.</p> <p>Control signaling: Indicated by red arrows.</p> <p>Power supply: Indicated by a red line.</p> <p>Excerpt, showing connection between host controller and WiFi device, from https://docs.microsoft.com/en-us/windows-hardware/design/device-experiences/wi-fi-power-management-for-modern-standby-platforms</p>

Claim 1	Accused Products
	 <p>The diagram illustrates the Windows networking stack and the architecture for managing a Wi-Fi device. It is divided into three horizontal layers: User-mode software, Kernel-mode software, and Hardware.</p> <ul style="list-style-type: none"> User-mode software: At the top, the Application layer (yellow) interacts with the Windows networking APIs (blue). Kernel-mode software: A dashed line separates the User-mode software from the Kernel-mode software. The Kernel-mode software layer contains the Windows networking config/control services (blue), the Wi-Fi device driver stack (PCI enumerated, ACPI filtered), and the Windows networking protocol drivers (blue). The Wi-Fi device driver stack includes the NDIS port driver, the Wi-Fi miniport driver (yellow), and the ACPI driver. The NDIS port driver and Wi-Fi miniport driver are shown in blue, while the ACPI driver is in yellow. The Windows networking config/control services interacts with the Wi-Fi device driver stack and the Windows networking protocol drivers. The Windows networking protocol drivers handle Pattern match offload requests. Hardware: The bottom layer represents the hardware components. It includes the System on a Chip (SoC) (orange), the PCIe host controller (orange), and the PCIe Wi-Fi device (orange). The PCIe host controller is connected to the PCIe Wi-Fi device via a blue arrow labeled Commands to Wi-Fi device. The PCIe Wi-Fi device is connected to the System on a Chip (SoC) via a blue arrow labeled D0/D3 transitions. The PCIe Wi-Fi device also has a red line labeled Power supply leading to a To PMIC or system power rail. <p>Legend:</p> <ul style="list-style-type: none"> Windows components: Represented by blue boxes. Partner components: Represented by yellow boxes. Hardware components: Represented by orange boxes. Data communication: Represented by blue lines. Control signaling: Represented by red lines. Power supply: Represented by red lines. <p>Excerpt: showing connection between host controller and WiFi device, from https://docs.microsoft.com/en-us/windows-hardware/design/device-experiences/wi-fi-power-management-for-modern-standby-platforms </p>

Claim 1	Accused Products
	<p>After all devices outside of the SoC, including communications devices, have been powered down, the host controllers on the SoC will be turned off. Almost every SoC has USB, I²C, GPIO, SDIO, and UART host controllers. Each of these components on the SoC must be turned off in order for the SoC to enter a low-power mode.</p> <p>The process for preparing the hardware for low-power during standby can be visualized as an upside-down pyramid, as shown in the following diagram. The lowest power is achieved when the whole SoC chip is powered down, but this can occur only after each set of devices above it in the pyramid has been powered down.</p>  <p>Devices outside SoC</p> <p>Network and radio devices</p> <p>On-SoC host controllers</p> <p>CPUs and GPUs</p> <p>Rest of SoC</p> <p>Excerpt describing connection between SOC and other devices, from https://docs.microsoft.com/en-us/windows-hardware/design/device-experiences/prepare-hardware-for-modern-standby</p>

Claim 1	Accused Products
[1c] a device coupled to the host controller;	<p>Each Accused Product includes a device coupled to the host controller.</p> <p>For example, the gram 15 includes a WiFi device coupled to the host controller.</p> <p><i>See, e.g.:</i></p>  <p>Screenshot, showing Wi-Fi device, from https://www.samsung.com/us/computing/galaxy-books/galaxy-book-flex/galaxy-book-flex-13-3-qled-512gb-storage-s-pen-included-np930qcg-k01us/</p>

Claim 1	Accused Products
	 <p>Legend</p> <ul style="list-style-type: none"> Windows components Partner components Hardware components <p>Data communication</p> <p>Control signaling</p> <p>Power supply</p> <p>Excerpt showing connection from Wi-Fi device to host controller from https://docs.microsoft.com/en-us/windows-hardware/design/device-experiences/wi-fi-power-management-for-modern-standby-platforms</p> <p>To support modern standby, Windows is designed to use the right network adapter at the right time for an Internet</p>

Claim 1	Accused Products
	<p>connection. All modern standby PCs have a Wi-Fi adapter, but some PCs also have a mobile broadband (MBB) adapter and/or a wired Ethernet adapter. During modern standby, Windows automatically connects to the best available network.</p> <p>***</p> <p>The Wi-Fi and MBB devices in a modern standby platform are expected to be highly autonomous. A key Wi-Fi device feature for modern standby is called network list offload (NLO). During modern standby, a Wi-Fi device that supports NLO can automatically connect to previously used Wi-Fi access points. NLO allows the Wi-Fi device to roam between previously used access points while the SoC remains in the low-power idle mode. As the user commutes between home and work, Windows automatically connects to Wi-Fi, thereby allowing the system to be already connected when the user presses the power button. The user no longer has to wait to connect to Wi-Fi. Instead, Wi-Fi is connected before the user turns on the system. Similarly, email is already downloaded, and connections to Skype and other communications services are automatically reconnected.</p> <p>Excerpt from https://docs.microsoft.com/en-us/windows-hardware/design/device-experiences/networking-power-management-for-modern-standby-platforms</p>

Claim 1	Accused Products
	<p>A Wi-Fi device in a modern standby platform must support several key power-management features to reduce power consumption by both the device and the platform as a whole.</p> <p>The first feature—called power save mode—allows the Wi-Fi device to reduce its power consumption while it remains connected to the access point. Power save mode reduces power consumption at the cost of increased data transfer latency. Power save mode is expected to always be enabled when the platform is running on battery power, except when low-latency connections are required (for example, for VOIP calls). For more information, see Wi-Fi Auto Power Save Mode.</p> <p>The second key power-management feature is pattern-match wake. This feature allows Windows to arm the Wi-Fi device to wake the System on a Chip (SoC) when the Wi-Fi device detects a network packet that matches a stored pattern. Pattern-match wake is operational only during modern standby. While pattern-match wake is enabled, the Wi-Fi device operates in a very low-power mode and listens for incoming data destined for specific system services or registered applications (for example, push notifications and email). Meanwhile, the other components in the hardware platform are in a low-power state. For more information, see Network Wake-Up Events.</p> <p>In addition, Wi-Fi devices in a modern standby platform must support the following run-time power-management features:</p>

Claim 1	Accused Products
	<ul style="list-style-type: none"> • Radio on/off state • Network list offload (NLO) • ARP/NS offload • D0 packet coalescing • Dynamic DTIM management • A set of Wi-Fi connectivity triggers, which include wake-on-AP-disconnect <p>Excerpts from https://docs.microsoft.com/en-us/windows-hardware/design/device-experiences/wi-fi-power-management-for-modern-standby-platforms</p>
<p>[1d] wherein the device is electrically disconnected from the host controller if the device is not in an active state; and</p>	<p>In each Accused Product, the device is electrically disconnected from the host controller if the device is not in an active state.</p> <p>For example, in the case where the claimed device is a Wi-Fi device, the device is not in an active state when the system enters Modern Standby idle mode. When the device is in this non-active state, the Windows 10 Modern Standby software electrically disconnects the device from the host controller, for example by configuring the device and/or host controller not to communicate with the SoC except for wake interrupts.</p> <p><i>See, e.g.:</i></p>

Claim 1	Accused Products
	<p>Transitioning from active to idle</p> <p>Windows transitions the SoC platform from active mode to idle mode after all software activity is stopped and the devices on and off the SoC chip have entered low-power states.</p> <p>The SoC transitions between idle and active modes during modern standby based on the triggers explained in the previous section. Whenever the SoC is active, Windows will aggressively attempt to return it to the idle (DRIPS) state to reduce power consumption.</p> <p>Note During modern standby, the SoC can be idle or active.</p> <p>The SoC will also be active when the system exits modern standby in response to user input, such as a power button press. The SoC will be put back in the idle (DRIPS) state automatically when all of the following conditions are true:</p> <ul style="list-style-type: none"> • All devices outside of the SoC have been powered down. • All network and radio devices have entered their low-power state to wait for interesting network packets or wake interrupts. • All host controllers on the SoC have been powered down. • All app background tasks have completed. • All CPU and GPU activity has stopped and all CPUs are idle. <p>Almost all modern standby power problems are related to making sure all of these five conditions have been met. Windows includes a built-in diagnostic tool called SleepStudy to help discover which of these conditions is not being met when the SoC is prevented from entering the idle (DRIPS) state.</p> <p>Excerpt, describing electrical disconnection, from https://docs.microsoft.com/en-us/windows-hardware/design/device-experiences/transitioning-between-idle-and-active-states</p>

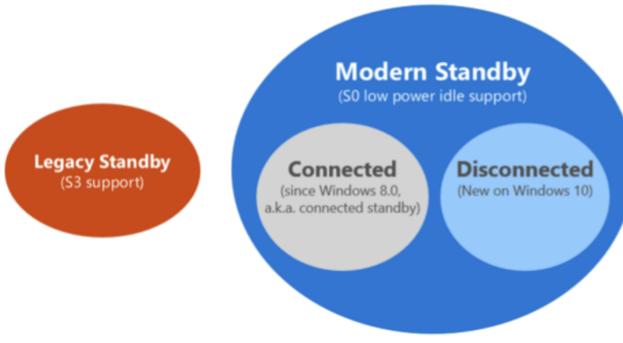
	<p><i>Connected sleep</i></p> <p>The Wi-Fi device is connected to the access point, but the remainder of the platform is in a very low-power state. Pattern-match wake is enabled so that the Wi-Fi device wakes the SoC on a specific set of incoming network packets.</p> <p>Transition mechanism:</p> <ul style="list-style-type: none">• Before the Wi-Fi device leaves D0, NDIS will send an OID_PM_ADD_WOL_PATTERN request to instruct the Wi-Fi miniport driver to add wake-on-LAN patterns.• To instruct the Wi-Fi miniport driver to enable pattern-match wake, NDIS will send an OID_PM_PARAMETERS request.• NDIS will send an OID_PNP_SET_POWER request with an NDIS_DEVICE_POWER_STATE value of NdisDeviceStateD2 (for SDIO) or NdisDeviceStateD3 (for PCIe). <p><i>Disconnected sleep</i></p> <p>The Wi-Fi device is powered but is not connected to an access point, because no preferred access point is within range. The remainder of the platform is in a very low-power state. Pattern-match wake is enabled and the Network Offload List is plumbed to the Wi-Fi device. The Wi-Fi device uses the Network Offload List to periodically scan for preferred networks to connect to.</p> <p>Transition mechanism:</p> <ul style="list-style-type: none">• The Wi-Fi device uses the network offload list to periodically scan for preferred networks to connect to.• If a matching network is found during these periodic scans, the Wi-Fi device will wake the SoC.
--	---

	<p><i>Radio off</i></p> <p>The Wi-Fi device still has power applied, but the radio (RF components) has been powered down.</p> <p>Transition mechanism:</p> <ul style="list-style-type: none">• While in D0, NDIS will send an OID_DOT11_NIC_POWER_STATE request with a value of FALSE, indicating the radio should be powered off. <p><i>Device powered down</i></p> <p>The Wi-Fi device has been completely powered down.</p> <p>Transition mechanism:</p> <ul style="list-style-type: none">• NDIS will send an OID_PNP_SET_POWER request with an NDIS_DEVICE_POWER_STATE value of NdisDeviceStateD3.• If the Wi-Fi device is connected to SDIO or PCIe, the system ACPI firmware will remove power from or reset the Wi-Fi device by using a GPIO line from the SoC to the Wi-Fi device.• If the Wi-Fi device is integrated into the SoC, the system firmware is responsible for powering off or resetting the Wi-Fi device by using a proprietary mechanism.
--	--

Claim 1	Accused Products
	<p>Pattern-match wake</p> <p>Windows requires all networking devices in a modern standby platform to support pattern-match wake. This feature enables the network device to monitor the network connection for interesting packets (by comparing incoming packets to patterns stored in the device) while the SoC or chipset is powered off, and to wake the main processor on the SoC or chipset when a matching pattern is detected. This ability to offload Wi-Fi processing from the main processor is key to achieving low-power operation in modern standby. Before the platform enters modern standby and the Wi-Fi device transitions to the connected-sleep (D2 for SDIO; D3 for PCIe) mode, Windows sends the Wi-Fi miniport driver a collection of interesting patterns that the Wi-Fi device must watch for.</p> <p>Pattern-match wake is the key enabling feature for modern standby. Pattern-match wake is enabled when the Wi-Fi device is in connected-sleep (D2/D3) mode. Before the Wi-Fi device enters this mode, Windows instructs the device to ignore all network packets except for those that match a specified set of patterns. Windows builds the matching patterns based on the services and applications (for example, push notifications and email) that are currently in use. Offloading pattern matching to the Wi-Fi device allows the rest of the hardware platform either to be turned off or to operate in a low-power mode. Meanwhile, the Wi-Fi device stays turned on, waiting for important incoming network traffic.</p> <p>Excerpts from https://docs.microsoft.com/en-us/windows-hardware/design/device-experiences/wi-fi-power-management-for-modern-standby-platforms</p>

Claim 1	Accused Products
	<h2>Connected vs. disconnected standby</h2> <p>Connected and disconnected Modern Standby states transitions follow a very similar code flow. The biggest difference is maintaining network connection. When a system goes into disconnected Standby state, it enters DRIPS, just like with the connected state. The primary difference is that it stops Wi-Fi traffic if a NIC that is not capable of offload support is detected. When a system enters disconnected standby, no patterns are configured for wake across LAN, Wi-Fi, or Mobile broadband. A media disconnect is sent to LAN, Wi-Fi and Mobile Broadband parts.</p> <p>Excerpts from https://docs.microsoft.com/en-us/windows-hardware/design/device-experiences/modern-standby-network-connectivity</p>
<p>[1e] wherein the device being electrically disconnected from the host controller causes an appearance to the host controller that the device is not coupled to the host controller; and</p>	<p>In each Accused Product, the device being electrically disconnected from the host controller causes an appearance to the host controller that the device is not coupled to the host controller.</p> <p>For example, under the Windows S0 low-power idle mode (“Modern Standby”), the host controller indicates that the network is disconnected. For another example, the Wi-Fi device enters an inactive state.</p> <p><i>See, e.g.:</i></p>

Claim 1	Accused Products
	<h2>Technical differences</h2> <p>When in the lowest power state, systems may look very similar to systems in the S3 state—processors are powered off and memory is in self-refresh. The difference is in the path of how it enters and exits low power state. For S3 systems, the system is either active or in S3. For Modern Standby, the transition from the active to the low power state is a series of steps to lower power consumption. Components are powered down when they are not in use. So, the transition into and out of a lower power state is much quicker on a Modern Standby system than on an S3 system. This design also helps with the speed of entry and exit from Standby as it doesn't require firmware interactions.</p> <p>Excerpts from https://docs.microsoft.com/en-us/windows-hardware/design/device-experiences/modern-standby-vs-s3</p>

Claim 1	Accused Products
	 <p>The diagram shows four modes of operation:</p> <ul style="list-style-type: none"> Legacy Standby (S3 support): Represented by an orange circle. Modern Standby (S0 low power idle support): Represented by a large blue circle containing two smaller circles. <ul style="list-style-type: none"> Connected (since Windows 8.0, a.k.a. connected standby): Represented by a grey circle. Disconnected (New on Windows 10): Represented by a light blue circle. <p>The text below the diagram provides context for the Modern Standby modes:</p> <p>The above figure illustrates the relationships between the models and how Modern Standby systems can either stay connected to or disconnect from the network while in standby.</p> <p>On any Modern Standby system, the system remains in S0 while in standby, allowing the following scenarios to work:</p> <ul style="list-style-type: none"> • Background activity • Faster resume from a low power state <p>On systems that can stay connected while in standby, wakes based on specific network patterns may also be set by the operating system to enable apps to receive the latest content such as incoming email, VoIP calls, or news articles.</p> <p>Excerpts from https://docs.microsoft.com/en-us/windows-hardware/design/device-experiences/modern-standby-vs-s3</p>

Claim 1	Accused Products
	<p>The Modern Standby user experience is designed to model that of a cellular phone. When users finish using their phones, they press the system power button and the cell phone enters sleep mode. The phone remains asleep until the user presses the power button again, or a phone call, email, or instant message is received.</p> <p>Similarly, when a PC is in Modern Standby, it looks and feels off—the screen is blanked, the system has no visible LED indicators, and there is no acoustic noise. However, a PC in Modern Standby remains on and connected to the Internet, just as the cell phone remains connected to the cellular network. (The Modern Standby PC uses any available network connection—Wi-Fi, mobile broadband (MBB)/cellular, or wired Ethernet.) And the Modern Standby PC, connected or not, also has very long battery life in its screen-off state, just like a cell phone.</p> <p>Excerpts from https://docs.microsoft.com/en-us/windows-hardware/design/device-experiences/modern-standby-wake-sources</p>
[1f] wherein a sideband signal is used to signal the device to electrically reconnect after the device has been electrically disconnected.	<p>In each Accused Product, a sideband signal is used to signal the device to electrically reconnect after the device has been electrically disconnected.</p> <p>For example, when the system exits Modern Standby idle state, the processor and/or host controller send a sideband signal to the Wi-Fi device instructing the Wi-Fi device to re-enter active state. As another example, when the system exits Modern Standby idle state, the processor and/or host controller send a sideband signal to the display device instructing the display device to turn on or re-enter active state.</p> <p><i>See, e.g.:</i></p>

Transitioning between idle and active states

02/14/2020 • 3 minutes to read •   

During modern standby, Windows periodically transitions the SoC platform from idle mode to active mode to perform value adding activities such as receiving an email notification or performing a maintenance task. Additionally, on-demand transitions to active mode can occur in response to user inputs, interrupts from networking devices, and other hardware events.

After the SoC has entered the DRIPS state, the platform will typically be in its lowest power mode. In modern standby, the platform will switch between this idle (DRIPS) mode and the active mode periodically. Low power is achieved in modern standby by spending the majority of the time in the DRIPS mode and very small amounts of time in the active mode.

Claim 1	Accused Products
	<p>Transitioning from idle to active</p> <p>Windows will transition the SoC to the active mode for the following reasons:</p> <ul style="list-style-type: none"> • An on-demand wake event due to an interrupt from a networking device (Wi-Fi, mobile broadband, or Ethernet). The networking device generates an interrupt when a matching WoL pattern is detected or when the media or connection state changes. The interrupt wakes the SoC using an out-of-band GPIO, in-band USB resume signaling, or PCI PME interrupt. • An on-demand wake event due to a platform device interrupt from a source such as the battery fuel gauge, battery charge controller, or thermal sensors. • An on-demand wake event due to an interrupt caused by user input at a keyboard, touchpad, external USB peripheral, power button, or lid switch. <p>As soon as the SoC wakes and the platform exits the DRIPS state, the CPUs start running code again. However, the screen stays powered off unless the interrupt was a result of user input or a power source change. The driver for the device that causes the interrupt is responsible for reporting user presence to the OS if the interrupt corresponds to user input. The Windows power manager determines whether the screen should be powered on; the system designer needs only to ensure that the interrupt from each device is connected and programmed to wake the SoC from the DRIPS state.</p> <p>System designers must work closely with the SoC vendor to connect device interrupts for SoC wakeup. Some SoC devices require device interrupts to be connected to special "wake-capable" inputs or configured for wake using system firmware.</p> <p>Excerpts, describing transition to active state, from https://docs.microsoft.com/en-us/windows-hardware/design/device-experiences/transitioning-between-idle-and-active-states</p>

Claim 1	Accused Products
	<p>All wake sources in the Modern Standby PC are expected to be capable of waking the SoC from its deepest idle power state. Some wake sources should be capable of generating a wake signal for an event that would cause the system display to turn on. The difference between waking the SoC and turning on the display is central to delivering the Modern Standby user experience. The following rules govern platform wake behavior:</p> <p>Excerpt, describing transition to active state, from https://docs.microsoft.com/en-us/windows-hardware/design/device-experiences/modern-standby-wake-sources</p>

Claim 14

Claim 14	Accused Products
[14pre] A method, comprising:	<p>To the extent the preamble is limiting, each Accused Product performs the claimed method.</p> <p><i>See supra</i> claim element [1pre].</p>
[14a] detecting whether a device coupled to a host controller is in an active state;	<p>Each Accused Product performs detecting whether a device coupled to a host controller is in an active state.</p> <p>For example, the device is detected to be in an active state when the system is not in Modern Standby idle mode.</p> <p><i>See, e.g.:</i></p>

Claim 14	Accused Products
	<p>Transitioning from active to idle</p> <p>Windows transitions the SoC platform from active mode to idle mode after all software activity is stopped and the devices on and off the SoC chip have entered low-power states.</p> <p>The SoC transitions between idle and active modes during modern standby based on the triggers explained in the previous section. Whenever the SoC is active, Windows will aggressively attempt to return it to the idle (DRIPS) state to reduce power consumption.</p> <p>Note During modern standby, the SoC can be idle or active.</p> <p>The SoC will also be active when the system exits modern standby in response to user input, such as a power button press. The SoC will be put back in the idle (DRIPS) state automatically when all of the following conditions are true:</p> <ul style="list-style-type: none"> • All devices outside of the SoC have been powered down. • All network and radio devices have entered their low-power state to wait for interesting network packets or wake interrupts. • All host controllers on the SoC have been powered down. • All app background tasks have completed. • All CPU and GPU activity has stopped and all CPUs are idle. <p>Almost all modern standby power problems are related to making sure all of these five conditions have been met. Windows includes a built-in diagnostic tool called SleepStudy to help discover which of these conditions is not being met when the SoC is prevented from entering the idle (DRIPS) state.</p> <p>Excerpt from https://docs.microsoft.com/en-us/windows-hardware/design/device-experiences/transitioning-between-idle-and-active-states</p>

	<p><i>Connected sleep</i></p> <p>The Wi-Fi device is connected to the access point, but the remainder of the platform is in a very low-power state. Pattern-match wake is enabled so that the Wi-Fi device wakes the SoC on a specific set of incoming network packets.</p> <p>Transition mechanism:</p> <ul style="list-style-type: none">• Before the Wi-Fi device leaves D0, NDIS will send an OID_PM_ADD_WOL_PATTERN request to instruct the Wi-Fi miniport driver to add wake-on-LAN patterns.• To instruct the Wi-Fi miniport driver to enable pattern-match wake, NDIS will send an OID_PM_PARAMETERS request.• NDIS will send an OID_PNP_SET_POWER request with an NDIS_DEVICE_POWER_STATE value of NdisDeviceStateD2 (for SDIO) or NdisDeviceStateD3 (for PCIe). <p><i>Disconnected sleep</i></p> <p>The Wi-Fi device is powered but is not connected to an access point, because no preferred access point is within range. The remainder of the platform is in a very low-power state. Pattern-match wake is enabled and the Network Offload List is plumbed to the Wi-Fi device. The Wi-Fi device uses the Network Offload List to periodically scan for preferred networks to connect to.</p> <p>Transition mechanism:</p> <ul style="list-style-type: none">• The Wi-Fi device uses the network offload list to periodically scan for preferred networks to connect to.• If a matching network is found during these periodic scans, the Wi-Fi device will wake the SoC.
--	---

	<p><i>Radio off</i></p> <p>The Wi-Fi device still has power applied, but the radio (RF components) has been powered down.</p> <p>Transition mechanism:</p> <ul style="list-style-type: none">• While in D0, NDIS will send an OID_DOT11_NIC_POWER_STATE request with a value of FALSE, indicating the radio should be powered off. <p><i>Device powered down</i></p> <p>The Wi-Fi device has been completely powered down.</p> <p>Transition mechanism:</p> <ul style="list-style-type: none">• NDIS will send an OID_PNP_SET_POWER request with an NDIS_DEVICE_POWER_STATE value of NdisDeviceStateD3.• If the Wi-Fi device is connected to SDIO or PCIe, the system ACPI firmware will remove power from or reset the Wi-Fi device by using a GPIO line from the SoC to the Wi-Fi device.• If the Wi-Fi device is integrated into the SoC, the system firmware is responsible for powering off or resetting the Wi-Fi device by using a proprietary mechanism.
--	--

Claim 14	Accused Products
	<p>Pattern-match wake</p> <p>Windows requires all networking devices in a modern standby platform to support pattern-match wake. This feature enables the network device to monitor the network connection for interesting packets (by comparing incoming packets to patterns stored in the device) while the SoC or chipset is powered off, and to wake the main processor on the SoC or chipset when a matching pattern is detected. This ability to offload Wi-Fi processing from the main processor is key to achieving low-power operation in modern standby. Before the platform enters modern standby and the Wi-Fi device transitions to the connected-sleep (D2 for SDIO; D3 for PCIe) mode, Windows sends the Wi-Fi miniport driver a collection of interesting patterns that the Wi-Fi device must watch for.</p> <p>Pattern-match wake is the key enabling feature for modern standby. Pattern-match wake is enabled when the Wi-Fi device is in connected-sleep (D2/D3) mode. Before the Wi-Fi device enters this mode, Windows instructs the device to ignore all network packets except for those that match a specified set of patterns. Windows builds the matching patterns based on the services and applications (for example, push notifications and email) that are currently in use. Offloading pattern matching to the Wi-Fi device allows the rest of the hardware platform either to be turned off or to operate in a low-power mode. Meanwhile, the Wi-Fi device stays turned on, waiting for important incoming network traffic.</p> <p>Excerpts from https://docs.microsoft.com/en-us/windows-hardware/design/device-experiences/wi-fi-power-management-for-modern-standby-platforms</p>

Claim 14	Accused Products
	<h2>Connected vs. disconnected standby</h2> <p>Connected and disconnected Modern Standby states transitions follow a very similar code flow. The biggest difference is maintaining network connection. When a system goes into disconnected Standby state, it enters DRIPS, just like with the connected state. The primary difference is that it stops Wi-Fi traffic if a NIC that is not capable of offload support is detected. When a system enters disconnected standby, no patterns are configured for wake across LAN, Wi-Fi, or Mobile broadband. A media disconnect is sent to LAN, Wi-Fi and Mobile Broadband parts.</p> <p>Excerpts from https://docs.microsoft.com/en-us/windows-hardware/design/device-experiences/modern-standby-network-connectivity</p>
<p>[14b] if the device is not in an active state, electrically disconnecting the device from a host controller, wherein electrically disconnecting the device from the host controller causes an appearance to the host controller that the device is not coupled to the host controller;</p>	<p>Each Accused Product performs if the device is not in an active state, electrically disconnecting the device from a host controller, wherein electrically disconnecting the device from the host controller causes an appearance to the host controller that the device is not coupled to the host controller.</p> <p><i>See supra</i> claim elements [1d] and [1e].</p>
<p>[14c] if the device is in an active state, maintaining an electrical connection between the device and the host controller; and</p>	<p>Each Accused Product performs if the device is in an active state, maintaining an electrical connection between the device and the host controller.</p> <p>For example, when the system is not in Modern Standby idle mode, the system does not electrically disconnect the device from the host controller.</p> <p><i>See supra</i> claim element [1d].</p>
<p>[14d] electrically reconnecting the device using a sideband signal after the device has been electrically disconnected.</p>	<p>Each Accused Product performs electrically reconnecting the device using a sideband signal after the device has been electrically disconnected.</p>

Claim 14	Accused Products
	<i>See supra</i> claim element [1f].

Claim 45

Claim 45	Accused Products
[45pre] A computer accessible memory medium that stores program instructions, wherein the program instructions are executable by a processor to:	Each Accused Product includes a computer accessible memory medium that stores program instructions, wherein the program instructions are executable by a processor to perform the claimed steps. <i>See supra</i> claim element [1pre].
[45a] detect whether a device coupled to a host controller is in an active state;	Each Accused Product contains program instructions executable to detect whether a device coupled to a host controller is in an active state. <i>See supra</i> claim element [14a].
[45b] if the device is not in an active state, electrically disconnect the device from a host controller, wherein electrically disconnecting the device from the host controller causes an appearance to the host controller that a device is not coupled to the host controller; and	Each Accused Product contains program instructions executable to if the device is not in an active state, electrically disconnect the device from a host controller, wherein electrically disconnecting the device from the host controller causes an appearance to the host controller that a device is not coupled to the host controller. <i>See supra</i> claim elements [1d] and [1e].
[45c] if the device is in an active state, maintain an electrical connection between the device and the host controller; and	Each Accused Product contains program instructions executable to if the device is in an active state, maintain an electrical connection between the device and the host controller. <i>See supra</i> claim element [14c].

Claim 45	Accused Products
<p>[45d] electrically reconnect the device using a sideband signal after the device has been electrically disconnected.</p>	<p>Each Accused Product contains program instructions executable to electrically reconnect the device using a sideband signal after the device has been electrically disconnected.</p> <p><i>See supra</i> claim element [1f].</p>

Claim 57

Claim 57	Accused Products
<p>[57pre] A system, comprising:</p>	<p>To the extent the preamble is limiting, each Accused Product comprises the claimed system.</p> <p><i>See supra</i> claim element [1pre].</p>
<p>[57a] a processor;</p>	<p>Each Accused Product includes a processor.</p> <p><i>See supra</i> claim element [1a].</p>
<p>[57b] a host controller coupled to the processor; and</p>	<p>Each Accused Product comprises a host controller coupled to the processor.</p> <p><i>See supra</i> claim element [1b].</p>
<p>[57c] a device coupled to the host controller;</p>	<p>Each Accused Product includes a device coupled to the host controller.</p> <p><i>See supra</i> claim element [1c].</p>
<p>[57d] wherein the device is electrically disconnected from the host controller if the device is not in an active state; and</p>	<p>In each Accused Product, the device is electrically disconnected from the host controller if the device is not in an active state.</p> <p><i>See supra</i> claim element [1d].</p>
<p>[57e] wherein a sideband signal is used to signal the device to electrically reconnect after the device has been electrically disconnected.</p>	<p>In each Accused Product, a sideband signal is used to signal the device to electrically reconnect after the device has been electrically disconnected.</p> <p><i>See supra</i> claim element [1f].</p>